Mediterranean Journal of Basic and Applied Sciences (MJBAS)
Volume 10, Issue 2, Pages 67-88, April-June 2026

Quantifying the Impact of Artificial Intelligence-Assisted Development Tools on the Computational Logic, Independent Problem-Solving Abilities, Academic Performance, and Coding Efficiency of Information Technology Students in Introductory Programming Education

Jade Marie P. Lagura[1*], Medalyn S. Amatiaga[2], Jame Cris Bugtong[3], Kezzel L. Balustre[4], Mickey B. Cabantac[5], Jefrey J. Dumalaoron[6], Desiree Joy S. Gamaya[7], Bjee F. Maestre[8], Chen T. Maniabo[9], July Hope E. Napiñas[10], Wengie S. Ordeniza[11], Kangel Cahnn B. Pollescas[12], Glaydel Ann O. Toledo[13] & Ginbert A. Fernandez[14]

[1-14]Department of Information Technology, College of Engineering and Technology, University of Science and Technology of Southern Philippines – Oroquieta Campus, Oroquieta City, Misamis Occidental, 7207, Philippines.
Corresponding Author (Jade Marie P. Lagura) Email: lagurajademarie078@gmail.com[*]

DOI: https://doi.org/10.46382/mjbas.2026.10205

Copyright © 2026 Jade Marie P. Lagura et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 23 February 2026 *Article Accepted: 24 April 2026* *Article Published: 26 April 2026*

ABSTRACT

This study assessed the effect of Artificial Intelligence (AI)-assisted programming tools on the coding performance of Information Technology (IT) students. In particular, the study sought to determine the impact of AI-assisted programming tools on students' programming performance, problem-solving abilities, motivation, and academic outcomes. The study utilized a descriptive-correlational research design, wherein 1,000 IT students from different institutions served as respondents. Mean and Pearson's r were used in analyzing the data gathered from the validated survey questionnaire.

The findings revealed that AI-assisted programming tools have a high impact on students' programming performance (M = 3.97). Students strongly agreed that these tools improved their understanding of programming concepts (M = 4.23), enhanced task completion speed (M = 4.28), and improved their programming scores (M = 4.63). In addition, there is a significant and strong positive relationship between AI-assisted programming tool usage and programming performance ($r = 0.62$, $p < 0.001$), thus rejecting the null hypothesis. This implies that as the use of AI-assisted programming tools increases, programming performance also improves. However, the study also found that students demonstrated only slight to moderate confidence in solving programming problems independently without AI support. The study concluded that AI-assisted programming tools are effective supplementary learning resources that improve students' efficiency, understanding, and academic performance in programming courses. Continuous and responsible use of these tools is necessary to maintain students' independent problem-solving and critical-thinking skills.

Keywords: Artificial Intelligence; AI-Assisted Programming Tools; Programming Performance; Computational Logic; Coding Efficiency; Information Technology Students; Introductory Programming Education; Problem-Solving Skills; Academic Performance; ChatGPT; Generative AI; Github Copilot.

1.0. Introduction

Based on the observation of its application in programming courses, programming has greatly benefited from Artificial Intelligence (AI) technology. With the introduction and the growth of AI-assisted programming tools such as GitHub, Copilot, and ChatGPT across the world, it is clear that they are widely employed by both teachers and students to improve code-writing and learning. These tools not only increase productivity but also provide instant feedback, which aids learners in identifying their mistakes and making corrections quickly enough. The usability of these AI applications is a significant step to address one of the main concerns regarding novices' possible frustration with implementing a learning tool that supports the generation of multiple solutions and promotes exploratory approaches. Consequently, learners have been able to carry out more experiments within a given shorter period, help them develop a deeper understanding of programming concepts and skills (Beltrán, 2025; Andleeb et al., 2025).

Generally, it is possible to observe the use of AI in education for many advantages, but it is not recommended to depend too much on this technology. Research has shown that students who rely too much on AI-generated code

gradually lose their problem-solving skills and independence, while core programming principles may also be misunderstood (Talandron-Felipe, 2024; Rahe, 2025). The AI system increases motivation because there is instantaneous feedback that allows the learners to make improvements in their study results. The self-confidence that learners have when using AI systems does not imply anything about the ability of these learners to learn programming ideas from Becker et al. (2023) and Long et al. (2025). This emphasizes applying a good balance between taking advantage of its positive aspects and encouraging nurturing thinking capacity so that they can learn how to program autonomously.

Locally, there is scant literature in the area regarding the influence of AI-assisted programming tools on the programming performance of first-year Information Technology (IT) students in classroom settings. The majority of current studies mostly concentrate on either general computer science students or online learning environments which represents a significant gap in AI technologies' impact on early Information Technology education practices and students' outcomes within a traditional academic framework.

The objective of this research is to investigate the effect of AI-assisted programming tools on programming performance, motivation, and learning outcomes among the first-year IT students, as well as the benefits and possible challenges of using AI in teaching programming to beginners. This will help educators realize how they can effectively integrate AI-assisted programming tools without compromising the meaningful learning process.

1.1. Statement of the Problem

The research examines how AI-assisted programming tools affect programming skills of Information Technology students in higher education. Educational programs require these tools, although their implementation creates several challenges which both students and teachers must overcome. The research specifically aims to investigate these particular research problems:

1) The extent to which AI-assisted programming tools influence the speed and accuracy of students' coding tasks.
2) The correlation between the frequency of AI-assisted programming tools usage and students' independent problem-solving performance and final grades.
3) The potential psychological impacts, such as changes in self-efficacy and the development of an "illusion of competence."

1.2. Hypotheses of the Study

Null Hypothesis (H_0): There is no significant relationship between the usage of AI-assisted programming tools and the programming performance of Information Technology students.

Alternative Hypothesis (H_1): There is a significant relationship between the usage of AI-assisted programming tools and the programming performance of Information Technology students.

1.3. Study Objectives

This study generally aims to determine the impact of Artificial Intelligence-assisted programming tools on the computational logic, programming performance, and coding efficiency of Information Technology students.

Specifically, this study aims to:

1. Determine the level of usage of AI-assisted programming tools among Information Technology students.
2. Identify the different purposes for using AI-assisted programming tools in programming-related tasks.
3. Assess the impact of AI-assisted programming tools on students' computational logic and coding efficiency.
4. Evaluate the influence of AI-assisted programming tools on students' academic performance and motivation in programming courses.
5. Determine the level of students' confidence in solving programming problems independently without AI assistance.
6. Examine the significant relationship between AI-assisted programming tool usage and programming performance among Information Technology students.

1.4. Significance of the Study

The study holds major importance for multiple stakeholders who make up the Information Technology education system:

- Students: The results of the study will show students how to use AI-assisted programming tools as learning aids which should be used together with their existing study methods to develop their skills.
- Educators: The research shows educators how to use AI in classrooms by providing evidence-based methods which create teaching approaches that develop critical thinking skills through AI usage.
- Curriculum Designers: The research findings show Information Technology curriculum designers about needed updates which require them to add "AI literacy" and create assessment standards for measuring student performance in AI-based learning environments.
- Educational Institutions: The research will support educational institutions in developing their policies and guidelines which define appropriate AI programming tool usage in technical and programming courses.
- Future Researchers: The study provides research material for future studies that investigate AI-assisted programming tools and their impact on programming performance and AI educational applications. Future researchers may also use the findings to further examine the long-term effects of AI-assisted programming tools on students' learning and independent problem-solving skills.

1.5. Scope and Delimitations

The study is focused on IT students enrolled in introductory programming courses during the academic year 2025-2026. The study investigates how people use ChatGPT and GitHub Copilot as their generative AI tools. The research includes three quantitative performance metrics which are grades and task completion time and code quality and the study also examines two qualitative perceptions which are motivation and self-efficacy.

The study only focuses on beginner programmers with little to no professional experience. It does not cover advanced IT students or professional developers. Furthermore, the study does not evaluate the internal algorithms of the AI tools themselves but rather their impact on student outcomes.

1.6. Definition of Terms

- AI-assisted Programming Tools: Generative AI platforms designed to help users in writing, debugging, and explaining code functions (e.g., GitHub Copilot, ChatGPT). These tools operate through Generative AI technology.
- Programming Performance: This metric evaluates programming skills through three distinct criteria which include measuring code accuracy and tracking the time taken to finish tasks and checking compliance with established programming guidelines.
- Novice Programmer: A student in the first year of an IT or CS degree students without any professional software development experience.
- Illusion of Competence: The learner develops a complete mastery of the subject because they can produce results using the tool, which creates a cognitive bias known as the illusion of competence.
- Quasi-Experimental Design: Through their quasi-experimental design, researchers examine the impact of AI-assisted programming tools on classrooms by incorporating AI-assisted programming tools into classrooms and observing students' use of technology.

2.0. Literature Review

Empirical data confirms that programming assistance from artificial intelligence can positively impact the efficiency and quality of coding by novice programmers. According to the research conducted by Beltrán (2025), students manage to complete their programming assignments faster since ChatGPT provides them with coding support and performs their basic work. Based on the investigation carried out by Andleeb et al. (2025), students had a more successful code as AI presented them with better answers than could have been obtained by novice programmers alone.

It can be concluded that there is a widespread understanding in recent scientific studies that AI technologies significantly enhance the effectiveness of work performance. As indicated in a meta-analysis of 35 experimental studies, the use of AI-enabled learning led to the faster completion of tasks with greater grades among students (Alanazi et al., 2025). These data support previous research where students claimed that AI technologies gave them quick answers and corrected errors in their programs, hence boosting their confidence (Yılmaz and Yılmaz, 2023). The relationship between the application of AI technology and academic performance of students, however, appears not to be quite clear. While empirical studies have revealed that programming assignments performed using ChatGPT yielded better academic results, there were no statistically significant differences in grades among the two compared categories of students – users of AI technology and non-users (Sun et al., 2024). The successful students do not employ AI technologies at all, while unsuccessful ones use this type of resource for completing their academic assignments (Stoyanova et al., 2025).

Static code analysis provides us with evidence confirming the research findings. Wermelinger (2023) found that GitHub Copilot significantly helped novice programmers solve simple programming problems faster and with reduced coding effort. However, the study also emphasized that excessive dependence on AI-generated code may

limit students' ability to fully understand programming logic and syntax independently. As Haindl and Weinberger (2024) state in their study, beginner Java programmers who used the ChatGPT service had higher code quality in terms of compliance with coding standards, while the complexity of their codes was considerably lower than that of the control group participants. In the work environment where GitHub Copilot was utilized, students were able to finish tasks 35% faster than others, as reported by Shihab et al. (2025) while they passed 50% more tests than students who worked without the tool. The results from larger classroom studies provide a more detailed understanding of the research findings. Kosar et al. (2024) conducted a controlled experiment with 182 first-year students and found that ChatGPT users and non-users did not show any significant difference in their total course grades or midterm exam performance ($p > .70$). Immediate task efficiency does not always result in academic achievements that accumulate over time.

The main worry of educators is that students will lose their ability to solve problems because of AI technology. According to Xue et al. (2024), students who used ChatGPT in introductory programming courses demonstrated improved task completion efficiency and debugging performance. Nevertheless, the researchers also noted that some students became overly dependent on AI-generated solutions during programming activities. Rahe and Maalej (2025) showed that students use AI for solution generation instead of scaffolding because they fail to understand the logic behind AI-generated solutions. The "trial-and-error" method results in performance drops because users need to stop using the AI system.

Talandron-Felipe (2024) showed that students who depended on ChatGPT for all their answers faced major difficulties when they needed to solve problems by themselves. Johnson et al. (2024) conducted an experiment with novice Arduino programmers and found that participants who wrote codes without the aid of AI fared better in their posttest performance compared to those who employed the services of ChatGPT ($p = .03$, $d = 0.76$). In relation to the findings by Jošt et al. (2024), those who rely on Large Language Models (LLMs) for critical thinking processes like code creation and debugging are likely to earn poor grades at the end of their learning process. Beginners should engage in writing the syntax manually since doing so allows them to learn crucial information that AI can help them bypass.

Although AI tools help in the timely completion of tasks, researchers have different opinions regarding their effects on the long-term development of mental abilities. In some studies, conducted on this issue, it is seen that the Generative AI will play an important role in enhancing computational thinking skills and problem-solving ability (Matobobo et al., 2025). On the other hand, other results indicate a gap between task completion and concept learning; for example, using AI as an "oracle" that gives the whole answer rather than a "help seeking tool" (Amoozadeh et al., 2024). Students' tendency to accept a full answer provided by AI without comprehending the answer causes students to think that they have already mastered everything despite their lacking necessary information (Amoozadeh et al., 2024).

The influence of AI technologies on the psychological experiences of learners is profound. Mhlanganiso and Makota explained that AI-powered code assistant tools positively influence programming proficiency and learning outcomes by helping students better understand coding structures, syntax, and debugging procedures. According to

the findings of Gardella et al. (2024), the introduction of AI-based Integrated Development Environment (IDE) AIDE led to decreased psychological strain and frustration that allowed users to work on complicated projects for more extended periods of time. The analysis provided by Ghimire and Edwards (2024) suggests that access to AI increased learners' willingness to engage in complex projects because they acquired increased initial motivation to learn.

However, this increased confidence is often decoupled from actual ability. Becker et al. (2023) highlighted the risk of "overblown self-efficacy," where students assume that they are proficient at some concept simply because the AI's code turned out well. Long et al. (2025) asserted that for motivational factors to foster learning, AI needs to be implemented in a "reflective" way, meaning that students need to reflect upon and analyze the AI-produced code. The results of the self-programming condition in Johnson et al. (2024) demonstrated higher levels of self-efficacy precisely because students realized that they were successful thanks to their own performance.

Researchers require new teaching methods to decrease current risks. Vadaparty et al. (2024) highlighted that integrating Large Language Models (LLMs) into introductory programming instruction can improve students' access to coding assistance and learning support. The study further explained that AI-assisted learning environments may enhance engagement when properly supervised by educators. Although some believe that AI encourages reflective thinking through self-monitoring and self-validation (Adejumo et al., 2026), others contend that conventional teaching needs to shift towards "Reverse Bloom's Taxonomy" activities that compel students to think critically about AI's output (Stoyanova et al., 2025). Educators need to oversee AI usage because it should serve as an educational tool which helps students learn programming skills instead of replacing their need for such skills (Marmolejo-Ramos et al., 2026; Matobobo et al., 2025).

The ubiquity of AI is necessitating a total rethink of programming assessments. The study by Alanazi et al. (2025) discovered that traditional "at-home" coding assessments now fail to evaluate student skills because students can use AI tools without critical thinking. The researchers recommend adopting "AI-aware" evaluation methods as an alternative. The assessment methods which researchers proposed for implementation include code explanation tasks, debugging AI-generated code, and modifying existing logic to meet new requirements (Alanazi et al., 2025).

The researchers Amiri and Islam (2025) proved their method worked by creating an AI-based code assistant which required students to use AI tools for error correction. The study found that programming skills improved by 34% while the time needed for debugging work decreased by 59.3%. The Lyu (2025) study investigated the PAI system which showed that students who employed this learning method achieved the best academic results because they treated AI technology as a "teammate" instead of using it as a "search engine" for information retrieval.

Mhlanganiso and Makota explained that AI-powered code assistant tools positively influence programming proficiency and learning outcomes by helping students better understand coding structures, syntax, and debugging procedures. Current research shows that AI technology produces two opposing effects according to the "Janus-faced" model which enables educators to experience higher productivity levels through AI but puts their primary ability to acquire knowledge and develop independent thought at risk. The research field lacks sufficient longitudinal studies which follow IT students throughout their full academic year to determine whether students

will overcome their "illusion of competence" and achieve actual skill mastery. Furthermore, effectiveness of AI tools depends on two factors which are the user's previous knowledge and their ability to control their learning process according to the research findings of Shihab et al. (2025) and Tan et al. (2024).

Table 1. Summary of Literature on AI-Assisted Learning in Programming Education

Feature	AI-Assisted Learning	Traditional Learning (Without AI)	Impact on Students
Task Completion	Faster Coding Assistance: AI tools like ChatGPT provide instant coding suggestions and solutions.	Manual Problem Solving: Students complete tasks independently through trial and error.	AI reduces task completion time and increases productivity.
Code Quality	Improved Code Standards: AI-generated code often follows proper syntax and coding conventions.	Beginner-Level Coding: Codes may contain more errors and inefficient structures.	AI enhances code readability and correctness.
Academic Performance	Mixed Academic Results: Some studies found no significant improvement in grades despite AI use.	Independent Learning: Students rely on their own understanding and practice.	AI may improve outputs but not always long-term academic achievement.
Problem-Solving Skills	AI Dependency: Students may rely on AI-generated answers without understanding the logic.	Critical Thinking Development: Students analyze and solve problems independently.	Excessive AI use may weaken analytical and debugging skills.
Learning Experience	Reduced Stress and Frustration: AI provides immediate support and error correction.	Higher Cognitive Effort: Students spend more time debugging and understanding concepts.	AI improves confidence and motivation during programming tasks.
Self-Efficacy	Overconfidence Risk: Students may feel skilled because AI-generated code works successfully.	Skill-Based Confidence: Confidence develops through actual coding practice.	AI can create an “illusion of competence.”
Computational Thinking	Reflective AI Use: AI can support computational thinking when used as a learning partner.	Hands-On Learning: Students fully engage in syntax writing and logic building.	Guided AI use can enhance higher-order thinking skills.
Assessment Methods	AI-Aware Assessments: Includes debugging AI-generated code and code explanation tasks.	Traditional Coding Exams: Focuses on manual coding performance.	Educators must redesign assessments for the AI era.
Long-Term Skill Development	Depends on Usage Style: AI as a “teammate” improves learning outcomes.	Consistent Skill Mastery: Independent practice strengthens long-term retention.	Balanced AI use is necessary for sustainable learning.
Educational Recommendation	Teachers guide and monitor AI use to support meaningful learning.	Conventional Instruction: Emphasizes independent coding and practice.	AI should support not replace programming education.

3.0. Methodology

3.1. Research Design

The study adopted a quantitative descriptive-correlational design in order to assess the effects of AI-assisted programming tools on the programming performance of IT students. With regard to the descriptive design aspect, the study sought to assess the degree of programming performance, use of AI-assisted programming tools, and students' perceptions on the advantages and disadvantages of AI-assisted programming tools. On the other hand, the correlational design sought to determine whether there is any significance relationship between AI-assisted programming tools and IT students' programming performance.

Descriptive-correlational research design was used since it offered researchers the opportunity to measure variables and examine their relationship without manipulation. As mentioned earlier, in this research, AI-assisted programming tool usage was considered as the independent variable, while students' programming performance was the dependent variable. By using this design, the researchers were able to explore the relationship between different levels of use of AI-assisted programming tools and differences in programming performance of students. Besides, this design mimics the real-life situation where students use AI-assisted programming tools. This kind of research design is therefore effective in studying the influence of AI-assisted programming tools among students.

3.2. Location of the Study

The study involved Information Technology (IT) students from various institutions who were enrolled in IT-related programs. The research team conducted their study with multiple schools because this approach allowed them to collect student responses who studied at different school locations.

3.3. Respondents of the Study

The study participants included first-year to fourth-year college IT students who had experience with AI-assisted programming tools such as ChatGPT and GitHub Copilot. The researchers selected these participants because their active involvement in programming subjects provided different experience levels which made them suitable for studying how AI-assisted programming tools affect programming skills and problem-solving abilities and academic results. The study participants were selected through purposive sampling. Researchers used this method to identify people who met the research eligibility criteria. The researchers selected IT students from first year to their fourth year and had programming skills and AI coding tool knowledge.

3.4. Data Gathering Procedure

The researchers followed a systematic and organized procedure in gathering the data to ensure accuracy and reliability. The study team created a survey questionnaire which followed their research goals through its structured question format. The instrument tested student understanding of AI-assisted programming tool usage together with their programming performance and motivation and their dealt challenges. The subject experts or instructors evaluated the questionnaire items to verify their clarity and relevance and suitability for instrument validity assessment.

The researchers conducted the survey after validation through two different methods to achieve higher response rates. Students received a paper-based survey which was conducted during class time while they could choose to complete the Google Forms survey online from any location. The researchers provided time for respondents to complete the questionnaire while they maintained voluntary participation. The participants learned that their answers would remain confidential and be used only for academic research purposes.

The research team gathered all survey data which included both paper-based and online responses and processed it into a unified dataset. The research team conducted a thorough review of responses to confirm answer completion before they proceeded with statistical analysis. The systematic process which followed this method enabled the team to gather precise data which maintained consistent results that could be interpreted.

Table 2. Response Mode and Scoring Guide

Numerical Rating	Range	Level of Agreement	Level of Impact
5	4.21-5.00	Strongly Agree	Very High Impact
4	3.41-4.20	Agree	High Impact
3	2.61-3.40	Neutral	Moderate Impact
2	1.81-2.60	Disagree	Low Impact
1	1.00-1.80	Strongly Disagree	Very Low Impact

3.5. Research Instrument

The researchers used a structured survey questionnaire as their primary research tool to obtain quantitative data about AI-assisted programming tool usage and its effects on the programming skills of Information Technology students. The study developed the questionnaire to measure two key study variables which included AI-assisted programming tool usage as the independent variable and programming performance as the dependent variable.

The questionnaire contained four primary sections. Part I: section of the study collected demographic information about respondents through three different variables which included age and gender and academic year and programming experience. Part II: section of the study assessed how often and which types of AI-Assisted Programming Tools people used for specific purposes through their experience with AI tools including ChatGPT and GitHub Copilot. Part III: section of the study assessed how students rated their coding abilities by measuring their skills inaccuracy and task completion time and debugging skills and problem-solving ability. Part IV: Perceptions on AI-Assisted Programming Tool examined both the perceived benefits and challenges of using AI tools, including motivation, dependency, and the development of an illusion of competence.

The questionnaire included three question types which consisted of multiple-choice items, Likert scale statements, and frequency-based questions that participants answered from “Always” to “Never.” The Likert scale was used to measure the level of agreement of respondents with statements related to AI tool usage and programming performance, while frequency-based items determined how often students used AI tools in their programming tasks. The researchers developed the instrument with straightforward language which enables participants to comprehend the content and provide accurate answers.

The questionnaire underwent content validation by subject matter experts who assessed its items to determine their relevance and clarity and their connection to the research objectives. The researchers will execute a pilot test with participants who match the characteristics of the research target group in order to evaluate instrument reliability. The researchers will use Cronbach's Alpha to assess the internal consistency of the questionnaire which will result in acceptable results when the coefficient reaches 0.70 or higher. The structured design of the questionnaire enables researchers to collect precise and organized data which can undergo statistical analysis for research purpose validation.

Table 3 presents the results of the reliability analysis of the research instrument. The calculated Cronbach's Alpha values demonstrate that the instrument used in the study is valid and reliable.

Table 3. Reliability Analysis

Scale	Cronbach's Alpha	No. of Items
Programming Performance	0.803	7

3.6. Statistical Treatment of Data

Researchers used statistical techniques to analyze their collected data which helped them achieve valid research outcomes and answer their study questions. The research team started their analysis process by encoding and organizing all survey responses which included both paper-based and Google Forms data collection methods.

The demographic profile which included age, gender, and year level data of the respondents was presented through Frequency and Percentage while the study used this method to present all AI-assisted programming tool usage data. The researchers used this method to present their data in a structured way that allowed for simple interpretation of their findings.

Frequency and Percentage

The researchers used frequency and percentage to analyze the demographic characteristics of the respondents who participated in the study and their usage of AI-assisted programming tools. The researchers applied statistical methods to identify how often participants utilized AI-assisted programming tools which included ChatGPT and GitHub Copilot for their programming tasks.

$$Percentage = \frac{f}{N}\ x\ 100$$

Where:

- f = frequency
- N = total number of respondents

Mean

The study used Mean calculations to find the average participant responses which were based on their Likert scale ratings. This statistical tool is appropriate for measuring central tendency and researchers used it to measure

AI-assisted programming tool usage and programming performance and to assess how respondents viewed AI tool benefits and challenges.

The formula for the mean is:

$$\bar{X} = \frac{\Sigma x}{N}$$

Where:

$\bar{x}$ = mean

Σx = sum of all responses

N = total number of respondents

The Pearson Product-Moment Correlation Coefficient (Pearson r) was utilized to establish the variable relationships. The statistical test is suitable because it determines the strength and direction of the relationship between two continuous variables, namely AI-assisted programming tool usage as the independent variable and programming performance as the dependent variable.

The formula is:

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

Where:

r = Pearson correlation coefficient

x = AI-assisted programming tool usage score

y = programming performance score

N = number of respondents

The correlation results were interpreted as follows:

Table 4. Table Interpretation of Pearson Correlation Coefficient (r-value)

R Value	**Interpretation**
0.80 – 1.00	Very Strong Relationship
0.60 – 0.79	Strong Relationship
0.40 – 0.59	Moderate Relationship
0.20 – 0.39	Weak Relationship
0.00 – 0.19	Very Weak Relationship

The table presents that the researchers used statistical methods to reveal data patterns and trends and data relationships which they used to establish study results and recommendations. The researchers used these tools to create study results which showed measurable evidence that was supported by statistical analysis.

3.7. Ethical Considerations

The researchers conducted the study while strictly following all established ethical standards throughout the entire research process. The study allowed research participation through voluntary participation which required

participants to receive full study information before answering the questionnaire. Informed consent was obtained from all participants who understood their study role and their right to withdraw from the study at any time without facing any negative consequences.

The researchers protected respondent information through their implementation of strict confidentiality protection measures which included maintaining respondent anonymity. The research team did not collect any personal identification details that included names and student numbers and they handled all research responses with complete confidentiality. The researchers maintained exclusive access to data that they had collected through paper-based surveys and Google Forms which they stored securely. The researchers used all collected research data exclusively for academic and research activities.

The researchers conducted the study while ensuring that no potential risks or harm would occur to the study participants. The questions were developed to maintain their suitability for the study while treating participants with full respect and avoiding any intrusive effects on their personal space. The research team maintain respect and integrity and responsibility through their ethical conduct which protected respondent rights and well-being throughout the entire research process.

4.0. Results and Discussion

Profile of Respondents

The following data present the characteristics of the respondents involved in the study on the impact of AI-assisted programming tools on the programming performance of Information Technology students.

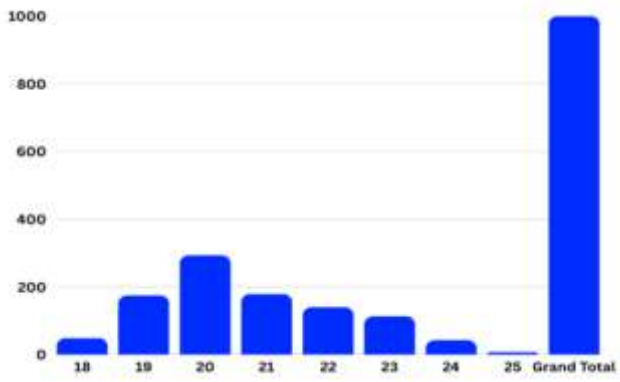

Figure 1. Distribution of respondents according to age among information technology students participating in the study on ai-assisted programming tools and programming performance.
Source: Survey Data Gathered by the Researchers (2026).

Figure 1 displays the age distribution of survey participants who took part in the researchers' study. The data show that respondents exist at various age levels which mostly match the standard college-age period.

The results show that the majority of respondents are concentrated between the ages of 18 to 24 with the highest number of responses coming from those who are 20 years old followed by individuals aged 21 and 19. A moderate number of respondents fall under the ages 22 and 23 while only a few respondents belong to the ages 24 and 25. The study mainly includes younger Information Technology students who show strong dedication to programming

work and AI-assisted programming tool usage according to this finding. The minimal representation of older age groups indicates that the adoption of AI-assisted programming tools is more prevalent among students within the early stages of their academic and programming experience.

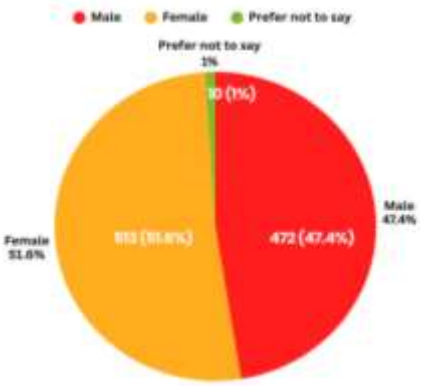

Figure 2. Gender distribution of information technology student respondents included in the study on AI-assisted programming tool usage and programming performance.
Source: Survey Data Gathered by the Researchers (2026).

The researchers conducted a survey which resulted in collected data that shows participant gender distribution in Figure 2. The study collected data from both male and female respondents while also including a small group of participants who chose to keep their gender identity private.

The survey results show that female respondents make up 51.6% of the total respondents while male respondents account for 47.4% and 1% of respondents chose to keep their gender identity private. The data shows that both male and female students participate in programming activities and they use AI coding tools for their work. The findings demonstrate that all genders use AI-assisted programming tools for programming work because these tools are accessible to all users.

Table 5. Purpose of Using AI-Assisted Programming Tools (Multiple Response)

Purpose	Frequency	Percentage
Code Generation	305	29.81%
Debugging Errors	274	26.78%
Understanding Programming Concepts	605	59.14%
Completing Assignments	374	36.56%

Table 5 shows the different purposes for which respondents use AI-assisted programming tools in their programming tasks. The results indicate that students use these tools for a variety of reasons.

The results show that most respondents use AI-assisted programming tools to learn programming concepts which accounts for 59.14% of their usage. Students use AI tools as their primary educational resource to enhance their understanding of programming subjects. The next most common use of AI-assisted programming tools is for

assignment completion which 36.56% of students use to meet their educational obligations. Students use AI tools for code generation which accounts for 29.81% of their usage and debugging errors which they use to resolve programming problems.

The research results demonstrate that AI-assisted programming tools function as a critical support system which helps Information Technology students improve their understanding of material while achieving higher work efficiency and completing their academic tasks. The study results show that AI tools play a crucial role in contemporary programming methods while improving students' educational experiences.

Table 6. Mean Distribution Showing the Impact of AI-Assisted Programming Tools on the Programming Performance of Information Technology Students.

Item Statements	Mean	Verbal Interpretation
My programming scores have improved after using AI-assisted programming tools	4.63	Strongly Agree
AI-assisted programming tools help me understand programming concepts	4.23	Strongly Agree
AI-assisted programming tools help me complete programming tasks more quickly	4.28	Strongly Agree
I rely on AI-assisted programming tools when solving programming problems	3.68	Agree
I can solve programming problems independently without AI-assisted programming tools	3.34	Neutral
Using AI-assisted programming tools affects my motivation to learn programming	3.70	Agree
AI-assisted programming tools influence my overall programming performance	3.95	Agree
Grand Mean	3.97	
Verbal Interpretation	High Impact	

Table 6 presents the impact assessment results on the use of AI-assisted programming tools in terms of programming performance. The data in Table 3 demonstrate that AI-assisted programming tools have a substantial effect on programming performance because the calculated grand mean score reached 3.97.

The participants showed strong agreement that their programming skills improved after they used AI-assisted programming tools ($\mu = 4.63$). They also strongly agreed that AI-assisted programming tools help them understand programming concepts ($\mu = 4.23$) and enable them to complete programming tasks more quickly ($\mu = 4.28$). The

research shows that AI-assisted programming tools provide major benefits to both programming learning and programming work efficiency.

The research results confirmed that participants use AI-assisted programming tools as their main programming problem-solving resource ($\mu = 3.68$), which shows that these tools function as essential assistance during their coding work. The participants confirmed that AI-assisted programming tools affect their programming skills ($\mu = 3.95$) together with their ability to study programming ($\mu = 3.70$), which shows that these tools positively impact both learning and performance. The participants showed their least ability to solve programming problems by themselves without AI-assisted programming tools which received a neutral rating of ($\mu = 3.34$). The survey showed that some participants who used AI-assisted programming tools benefited from these tools yet they remained doubtful about their ability to solve problems without using these tools. The research findings show that AI-assisted programming tools create major positive effects which enhance the programming skills of Information Technology students. The tools help users improve their understanding and academic performance while they create some dependency problems for users.

Table 7. Programming Performance Level

Performance Level	Frequency	Percentage
Low	69	6.90%
Very Low	92	9.20%
Moderate	724	72.40%
High	100	10.00%
Very High	15	1.50%
Total	1000	100%

Table 7 displays the programming skills assessment results which respondents provided about their own abilities. The study found that most respondents assessed their programming skills at Moderate level which corresponds to 72.40% of total participants who believe their abilities match the standard average. The remaining respondents assessed their abilities at High level (10.00%) and Low level (9.20%) while only 6.90% of participants selected Very Low and 1.50% selected Very High as their self-assessment categories. Overall, the findings suggest that most Information Technology students have a moderate level of programming performance, with only a few perceiving themselves at the highest level.

Table 8. Confidence in Solving Problems Without AI-Assisted Programming Tools

Confidence Level	Frequency	Percentage
Not Confident	200	20.0%
Slightly Confident	400	40.0%
Moderately Confident	400	40.0%
Total	1000	100%

Table 8 shows the level of confidence of respondents in solving programming problems without the use of AI-assisted programming tools. The results indicate that most respondents are either Slightly Confident (40.00%) or Moderately Confident (40.00%), while 20.00% reported being Not Confident. This implies that although students have some level of confidence in their programming abilities, many still depend on AI-assisted programming tools to assist them in solving programming tasks. Overall, the findings suggest that students are not fully confident working independently without AI support.

Table 9. Confidence in Solving Problems Without AI-Assisted Programming Tools

Grade Range	Frequency	Percentage
Below 70	4	0.40%
75–79	29	2.90%
80–84	179	17.90%
85–89	433	43.30%
90–94	325	32.50%
95 and above	30	3.00%
Total	1000	100%

Table 9 presents the most recent programming grades of the respondents after using AI-assisted programming tools. The results demonstrate that most respondents achieved grades between 85 and 89 which accounted for 43.30% of the total while 32.50% of respondents achieved grades between 90 and 94. A smaller percentage of respondents obtained grades within 80–84 (17.90%), while only a few fall under 75–79 (2.90%), 95 and above (3.00%), and below 70 (0.40%). The results demonstrate that students achieved satisfactory to high grades through their use of AI-assisted programming tools which resulted in better programming skills.

Table 10. Correlation Between AI-Assisted Programming Tools Usage and Programming Performance

Variables	r-value	Interpretation
AI-Assisted Programming Tool Usage and Programming Performance	0.62	Strong Positive Relationship

Table 10 shows how Information Technology students use AI-assisted programming tools which affects their programming abilities. The computed Pearson r-value of 0.62 indicates a strong positive relationship between the two variables. The study found that student programming performance improves when students use more AI-assisted programming tools. The result shows that AI-assisted programming tools help students reach better efficiency and understanding of programming concepts and they achieve higher academic results.

The findings support the assumption that AI-assisted programming tools have a significant impact on programming performance, making them effective as supplementary learning tools. However, while the relationship is strong, it does not imply that AI usage alone determines performance, as other factors may also influence students' outcomes.

5.0. Conclusion and Future Recommendations

5.1. Summary of Findings

This study examined the impact of Artificial Intelligence (AI)-assisted programming tools on the computational logic, programming performance, and coding efficiency of Information Technology students in introductory programming education.

A total of 1,000 student respondents participated in the study using a structured survey questionnaire. The data gathered were analyzed using frequency and percentage distribution, mean, and Pearson Product-Moment Correlation Coefficient (Pearson r).

The major findings of the study are summarized as follows:

Descriptive Results

- The majority of respondents belonged to the age group of 18 to 24 years old, indicating that most participants were actively engaged in programming-related academic activities.
- Most students utilized AI-assisted programming tools for understanding programming concepts, completing assignments, generating code, and debugging programming errors.
- Understanding programming concepts obtained the highest percentage among the identified purposes of AI-assisted programming tool usage.
- The overall impact of AI-assisted programming tools on programming performance obtained a grand mean of 3.97 interpreted as “High Impact.”
- Students strongly agreed that AI-assisted programming tools improved their understanding of programming concepts, increased task completion speed, and enhanced programming performance.
- Most respondents demonstrated only slight to moderate confidence in solving programming problems independently without AI assistance.

Descriptive Results

- The correlation analysis revealed a strong positive relationship between AI-assisted programming tool usage and programming performance.
- The computed Pearson r-value of 0.62 indicated that increased usage of AI-assisted programming tools was associated with better programming performance and coding efficiency among Information Technology students.
- The findings suggest that AI-assisted programming tools positively contribute to students’ academic performance, learning efficiency, and understanding of programming concepts.

Overall Findings

- AI-assisted programming tools served as effective supplementary learning resources for beginner programmers and Information Technology students.
- Although AI-assisted programming tools improved productivity and academic outcomes, excessive dependence on these tools may weaken students' independent problem-solving and critical-thinking abilities.
- The findings emphasize the importance of balancing AI-assisted learning with independent coding practice to maintain students' computational thinking and programming skills.

5.2. Conclusions

The research study results demonstrate that AI-assisted programming tools for students in Information Technology education improve their programming skills. The tools help students to work more efficiently while they learn programming concepts better. The research shows that students who depend on the tools will lose their ability to solve programming challenges independently. The findings demonstrate that AI-assisted programming tools improve programming performance; however, excessive dependence on these tools may hinder the development of independent programming knowledge. Students should use AI-assisted programming tools for learning because they are better learning tools yet students must use them in equal measure to sustain their critical thinking and problem-solving abilities.

5.3. Recommendations

Based on the conclusions of the study, the following recommendations are proposed:

1. Future researchers may investigate the long-term effects of Artificial Intelligence (AI)-assisted programming tools on students' independent problem-solving skills, computational thinking, and critical-thinking abilities.
2. Future studies may compare the effectiveness of different AI-assisted programming platforms such as ChatGPT, GitHub Copilot, Gemini, and other generative AI systems in programming education.
3. Educational institutions should develop clear policies, ethical guidelines, and responsible AI usage frameworks for programming-related academic activities.
4. Educators are encouraged to integrate AI-assisted programming tools as supplementary learning resources while still promoting independent coding practice and logical reasoning among students.
5. Future researchers may conduct experimental or longitudinal studies involving larger populations and different academic disciplines to obtain more comprehensive findings regarding AI-assisted learning technologies.
6. Students should continuously strengthen their manual coding, debugging, and analytical skills to avoid excessive dependence on AI-generated programming solutions.

Declaration

Source of Funding

This research did not benefit from grant from any non-profit, public or commercial funding agency.

Competing Interests Statement

The authors have declared that no competing financial, professional or personal interests exist.

Consent for publication

All the authors contributed to the manuscript and consented to the publication of this research work.

Authors' Contributions

All authors contributed equally to the conceptualization, data gathering, statistical analysis, interpretation of results, manuscript preparation, proofreading, and revision of the study. All authors approved the final manuscript prior to submission.

Informed Consent

Informed consent was obtained from all participants involved in the study before the conduct of data collection procedures.

Availability of data and material

Supplementary information such as the raw files of the data gathering and data analysis are available from the authors upon reasonable request.

Institutional Review Board Statement

Not Applicable.

Ethical Approval

Not Applicable.

Acknowledgments

We, researchers, would like to thank our Almighty Father for giving us enough knowledge and wisdom that we are able to finish our research paper despite all the difficulties we encountered along the way.

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this research paper.

Our greatest appreciation to Mr. Ginbert A. Fernandez has provided assistance to us through his ongoing support according to his role as our Quantitative Methods instructor. He has a valuable contribution in shaping the direction and quality of this research.

We are indebted to the participants who dedicated their time to help gather the essential information needed for our research. Their personal experiences and insights, which significantly improved the research findings and overall study conclusions.

Lastly, we would like to express our heartfelt appreciation to the families for their unwavering love, encouragement, and understanding during this research endeavor. They continuously supported us while they maintained their confidence in our abilities, which became our primary motivation throughout the research.

While it is impossible to acknowledge everyone who has played a role in this research, yet we want to express our thanks to all who participated in any research activities. Your assistance and support have been invaluable, and we are truly grateful for your involvement.

Declaration of Artificial Intelligence

The authors utilized artificial intelligence tools solely for grammar checking, language refinement, and formatting assistance during the preparation of this manuscript. All interpretations, analyses, conclusions, and final written content remain the sole responsibility of the authors.

References

[1] Gutiérrez Beltrán, E.J., & Martínez Arias, J.C. (2024). Mi Superpoder es la Programación: An educational resource that enables children and youth to learn programming skills. Entertainment Computing, 50: 103198. Retrieved February 18, 2026, from https://www.sciencedirect.com/science/article/pii/S0167642324001217.

[2] Alanazi, M., Soh, B., Samra, H.E., & Li, A. (2025). The influence of artificial intelligence tools on learning outcomes in computer programming: A systematic review and meta-analysis. Computers, 14(5): 185. https://doi.org/10.3390/computers14050185.

[3] Amiri, S. (2025). Enhancing Python programming education with an AI-powered code helper: Design, implementation, and impact. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5240568.

[4] Amiri, S.M.H., & Islam, M.R. (2025). Enhancing Python programming education with an AI-powered code helper: Design, implementation, and impact. Software Engineering, 11(1): 1–17. https://doi.org/10.11648/j.se.20251101.1.

[5] Andleeb, S., Kantorski, B., & Carver, J.C. (2025). ChatGPT in introductory programming: Counterbalanced evaluation of code quality, conceptual learning, and student perceptions. arXiv. https://arxiv.org/abs/2510.00946v1.

[6] Becker, B., Craig, M., Denny, P., & Keuning, H. (2023). Generative AI in introductory programming. ACM Computer Science Education. Retrieved from https://csed.acm.org/wp-content/uploads/2023/12/generative-ai-nov-2023-version.pdf.

[7] Can generative pre-trained transformers (GPT) pass assessments in higher education programming courses? (2023). https://doi.org/10.48550/arxiv.2303.09325.

[8] Gardella, N., Pettit, R., & Riggs, S.L. (2024). Performance, workload, emotion, and self-efficacy of novice programmers using AI code generation. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery. https://doi.org/10.1145/3649217.3653615.

[9] Ghimire, A., & Edwards, J. (2024). Coding with AI: How are tools like ChatGPT being used by students in foundational programming courses. In Lecture Notes in Computer Science, Springer. https://doi.org/10.1007/978-3-031-64299-9_20.

[10] Haindl, P., & Weinberger, G. (2024). Does ChatGPT improve coding abilities for beginning programmers to write better code? Results from static code analysis. Preprints. https://doi.org/10.20944/preprints202406.1151.v1.

[11] Kazemitabaar, M., Chow, J., Ma, C.K.T., Ericson, B., Weintrop, D., & Grossman, T. (2023). Studying the effect of AI code generators on supporting novice learners in introductory programming. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery. https://doi.org/10.1145/3544548.3580919.

[12] Kristensen, L., & Høyland, S. (n.d.). Introducing coding co-pilots in an introductory programming course—the beginning of the end? Retrieved from https://www.ntnu.no/ojs/index.php/nikt/article/view/6523.

[13] Long, X., Tan, X., Zhu, Y., Jiang, J., & Zhang, L. (n.d.). Understanding and enhancing CS students' interaction experience with AI coding assistant tools. Association for Computing Machinery. Retrieved from https://dl.acm.org/doi/10.1145/3785479.

[14] Maher, M.L., Tadimalla, S.Y., & Dhamani, D. (2023). An exploratory study on the impact of AI tools on the student experience in programming courses: An intersectional analysis approach. In Proceedings of the Frontiers in Education Conference, Pages 1–5, IEEE. https://doi.org/10.1109/fie58773.2023.10343037.

[15] Mäntymäki, V. (n.d.). AI for software engineering education: Evaluating AI-powered assistants for learning programming. LUT University. Retrieved from https://lutpub.lut.fi/handle/10024/170446.

[16] Mhlanganiso, T., & Makota, J. (n.d.). Enhancing programming proficiency: Evaluating the impact of AI-powered code assistant tools on learning outcomes. Retrieved from https://journals.zegu.ac.zw/index.php/o/article/download/411/302.

[17] Nikolaidis, N., Flamos, K., Gulati, K., Feitosa, D., Ampatzoglou, A., & Chatzigeorgiou, A. (2024). A comparison of the effectiveness of ChatGPT and Co-Pilot for generating quality Python code solutions. In Proceedings of the 2024 IEEE International Conference on Software Analysis, Evolution and Reengineering, Pages 93–101, IEEE. https://doi.org/10.1109/saner-c62648.2024.00018.

[18] Rahe, C., & Maalej, W. (2025). How do programming students use generative AI? Proceedings of the ACM. https://doi.org/10.1145/3715762.

[19] Shihab, M.I.H., Hundhausen, C., Tariq, A., Haque, S., & Mulanda, B. (2025). The effects of GitHub Copilot on computing students' programming effectiveness, efficiency, and processes in brownfield programming tasks. In Proceedings of the ACM Technical Symposium on Computer Science Education, Association for Computing Machinery. https://doi.org/10.1145/3702652.3744219.

[20] Sørli, C., & Aalberg, T. (n.d.). How and why novices use LLM-based tools to solve CS1 coding tasks. Retrieved from https://www.ntnu.no/ojs/index.php/nikt/article/view/6555.

[21] Sun, S.-K., Li, M., & Yang, B. (2025). Research on the application of AI code assistants in C language programming courses. In 2025 Conference on Science, Technology and Engineering, Pages 20–24, IEEE. https://doi.org/10.1109/cste64638.2025.11091939.

[22] Talandron-Felipe, M. (n.d.). Efficiency or understanding? Novice programmers' problem-solving with and without ChatGPT. Retrieved from https://library.apsce.net/index.php/icce/article/view/6065.

[23] Vadaparty, A., Zingaro, D., Smith, D.H., Padala, M., Alvarado, C., Benario, J.G., & Porter, L. (2024). CS1-LLM: Integrating LLMs into CS1 instruction. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery. https://doi.org/10.1145/3649217.3653584.

[24] Wermelinger, M. (2023). Using GitHub Copilot to solve simple programming problems. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023), Association for Computing Machinery. https://doi.org/10.1145/3545945.3569830.

[25] Xue, Y., Chen, H., Bai, G.R., Tairas, R., & Huang, Y. (2024). Does ChatGPT help with introductory programming? An experiment of students using ChatGPT in CS1. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24), Pages 331–341, Association for Computing Machinery. https://doi.org/10.1145/3639474.3640076.